**Informe Técnico**

**Explotación Controlada de Inyección SQL en DVWA y Documentación del Hallazgo**

**1. Título del Hallazgo**

**Vulnerabilidad de Inyección SQL (SQLi) en Módulo de Usuarios – DVWA (entorno controlado)**

**2. Descripción del Problema**

Durante una auditoría interna en el entorno de laboratorio **DVWA (Damn Vulnerable Web Application)**, se identificó que el módulo de usuarios es vulnerable a **SQL Injection**. La aplicación no implementa validación ni parametrización de consultas SQL, lo que permite inyectar código malicioso directamente en el parámetro id. Esto podría comprometer la base de datos, exfiltrar información sensible y, en un entorno real, escalar a un compromiso total del sistema.

**3. Evidencia Técnica**

**Herramientas utilizadas:**

* Navegador web (DVWA en http://localhost/dvwa)
* **Burp Suite** (interceptación y repetición de solicitudes)

**Payloads ejecutados:**

1. **Prueba booleana:**
2. 1' OR '1'='1

✔ Resultado: devolvió múltiples registros, confirmando inyección.

1. **Prueba UNION SELECT:**
2. 1' UNION SELECT null, database(), null --

✔ Resultado: se mostró el nombre de la base de datos activa.

**Evidencia (resumen visual):**

* Respuestas HTTP con resultados adicionales no autorizados.
* Aparición de errores SQL visibles al manipular los parámetros.

**4. Evaluación del Riesgo**

* **Clasificación:** **Crítico**
* **Impacto potencial:**
  + Exfiltración de credenciales y datos sensibles.
  + Modificación o eliminación de registros.
  + Posible escalada hacia control completo del servidor de base de datos.
* **Vector de ataque:** remoto, bajo esfuerzo de explotación.
* **Referencias:** OWASP Top 10 – A03:2021 Injection.
* **Severidad (CVSS v3.1):** 9.1 (Critical).

**5. Recomendaciones Técnicas**

1. Implementar **consultas parametrizadas (Prepared Statements)** en lugar de concatenación dinámica.
2. Aplicar **validación y saneamiento de entradas** (whitelisting, tipos de datos estrictos).
3. Configurar el sistema para **no mostrar errores SQL al cliente**.
4. Implementar un **WAF (Web Application Firewall)** para filtrar intentos de inyección.
5. Capacitar al equipo de desarrollo en **OWASP Top 10** y prácticas seguras de codificación.

**6. Referencias**

* OWASP Top 10 – A03:2021 Injection
* [NIST SP 800-115 – Technical Guide to Information Security Testing]
* [MITRE CVE Database – SQL Injection entries]

**7. Reflexión Final**

La inyección funcionó porque la aplicación concatenaba directamente la entrada del usuario en la consulta SQL, sin validación ni parametrización.  
Se habría evitado utilizando **prepared statements** y validaciones estrictas.  
Si este fallo estuviera en un sistema en producción, un atacante podría comprometer seriamente la base de datos, robar información sensible y afectar la continuidad del negocio.